UNIST
ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY
2007

Paper Review

Efficient and Robust LiDAR-Based End-to-End Navigation

Zhijian Liu, Alexander Amini, et al. ICRA 2021

Department of Computer Science, UNIST
UNIST brAIns, Hyeonggyu Kim

CONTENTS

1. Introduction
   - Problem statement

2. Methodology
   - Fast-LiDARNet
   - Hybrid Evidential Fusion

3. Experiments
   - Setup
   - Data augmentation
   - Training

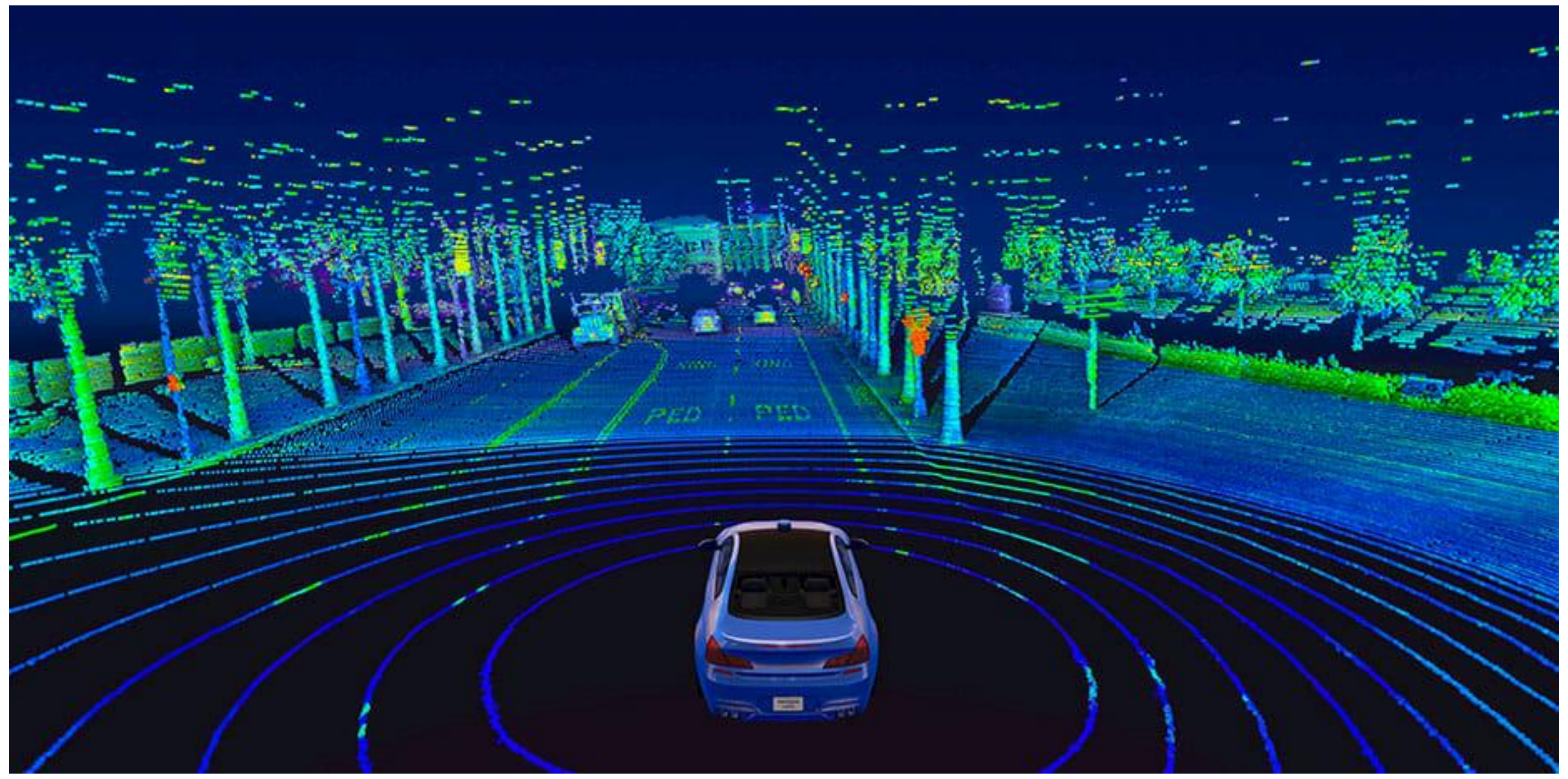
4. Result

5. Conclusion
   - Limitations

Problem statement 1

Efficiency: Computationally challenging

- Unordered structure and Large size
- SOTA 3D network
  : 14x more computation than ResNet
- 2D-based solution(e.g. bird's-eye view)
  : Loss of geometric information



https://velodynelidar.com/blog/guide-to-lidar-wavelengths/

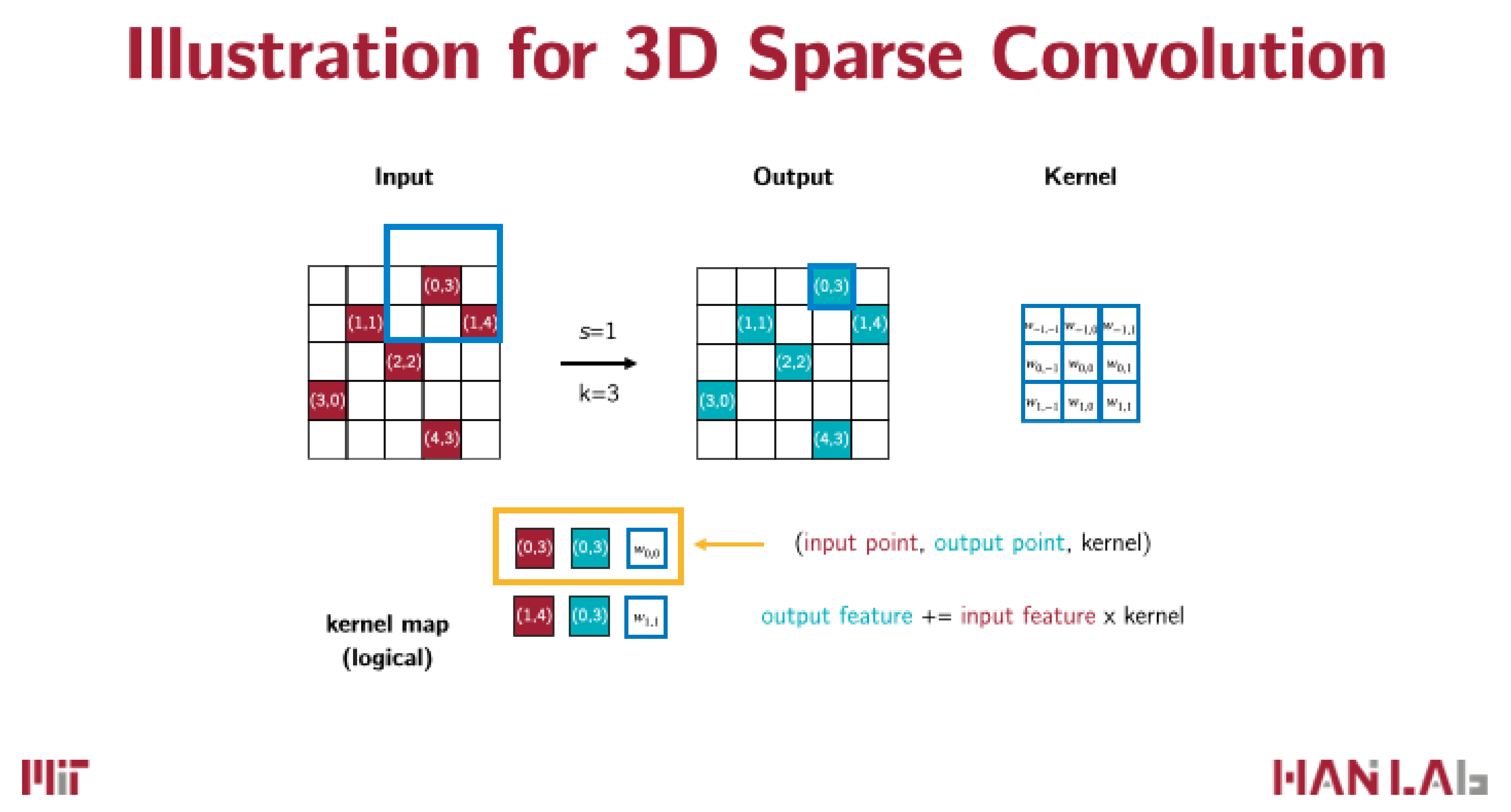
Introduction

Recent approach and Motivation

Usage of sparse convolution

- MinkowskiNet, PolarNet, SPVCNN

- Issue: Irregular memory access pattern



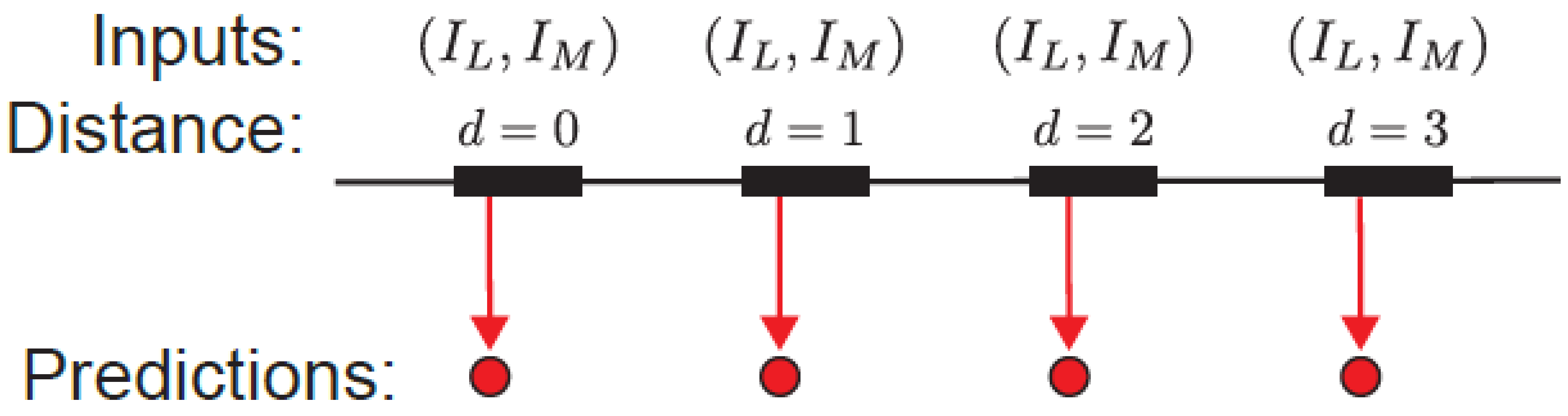
"Hardware-aware neural architectures are required."

https://hanlab.mit.edu/projects/spvnas/

Problem statement 2

Robustness: Instantaneous Reactive control

- High sensitivity to perturbations (e.g. noisy sensory inputs)



Introduction

Recent approach and Motivation

Usage of consecutive frames as an input

- Modeling recurrence or 4D convolutions is required
- Issue: Not efficient and Not effective in closed-loop control setting

"More efficient model which predict future control is needed."

Contributions

1. Efficiency

- Fast-LiDARNet, a hardware-aware neural architecture

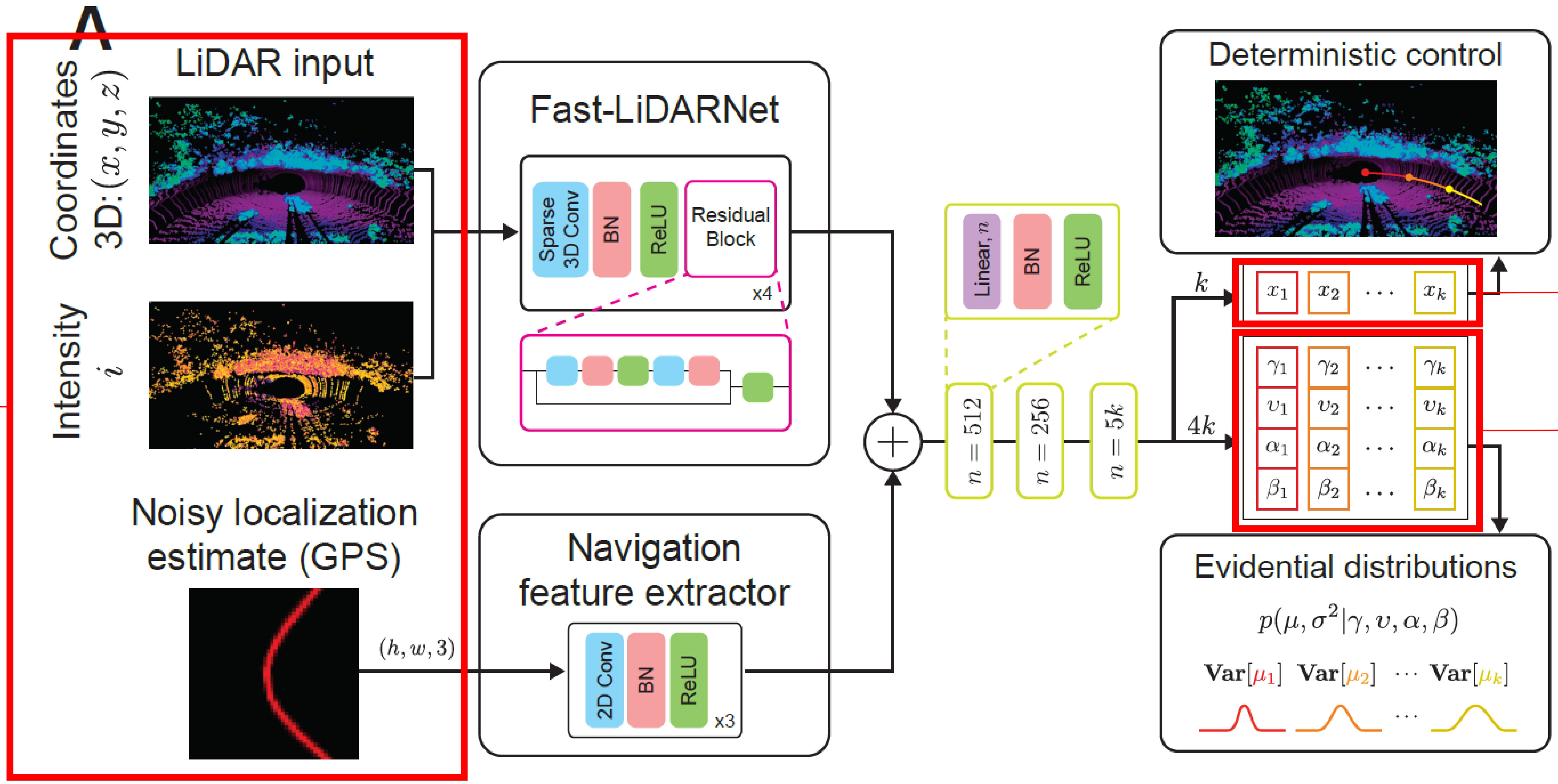
2. Robustness

- Hybrid Evidential Fusion, an algorithm learns prediction uncertainties and adaptively integrates predictions

Methodology

Overall architecture

Inputs



Curvature

Parameters for measuring uncertainty

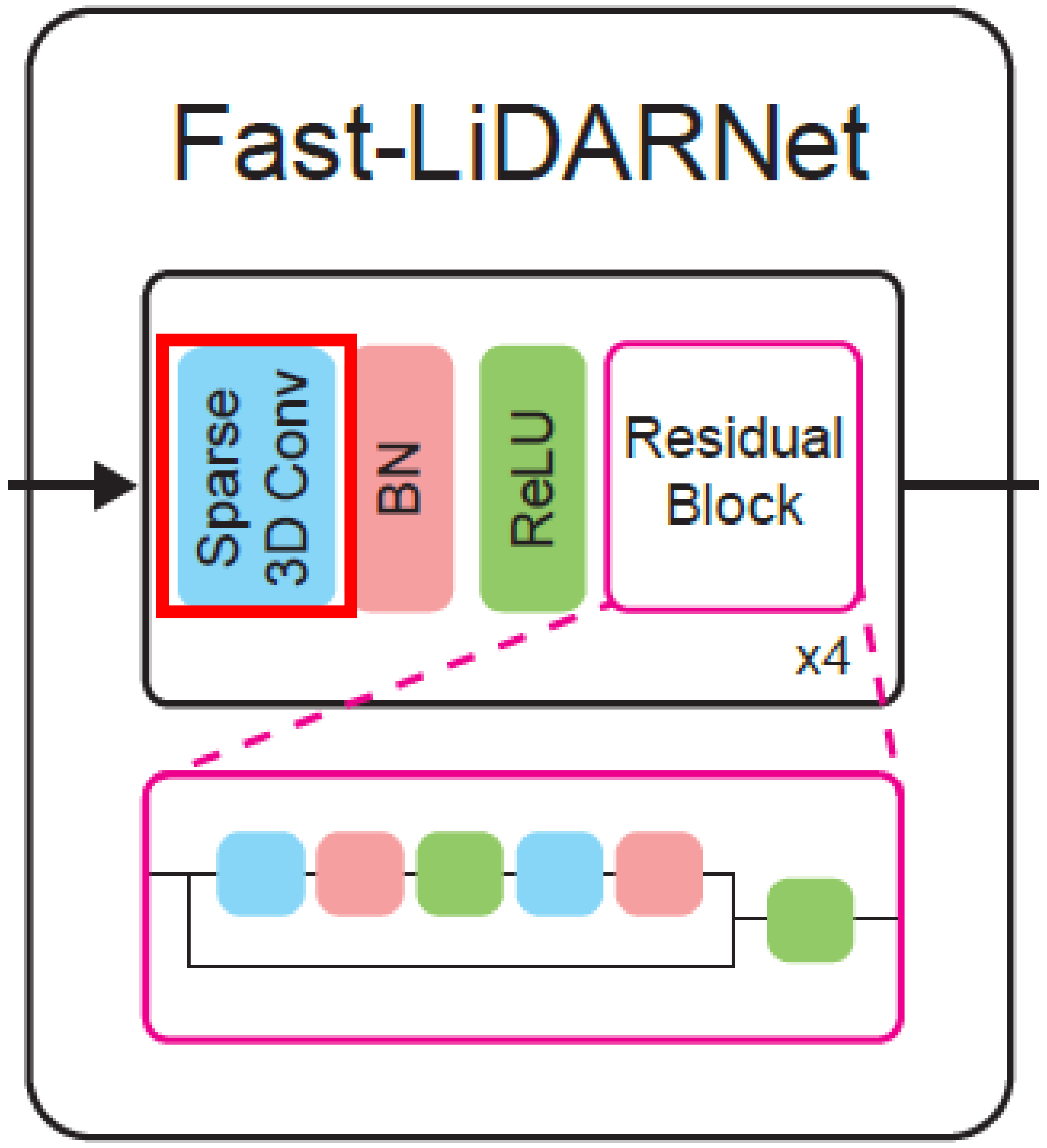
Methodology

Fast-LiDARNet

1. Sparse kernel optimization

- Index lookup of sparse convolution on GPUs
  : reduces latency of kernel map construction to 10%

- Preprocessing with gathering procedure(similar to im2col)
  : GEMM on large, regular matrix



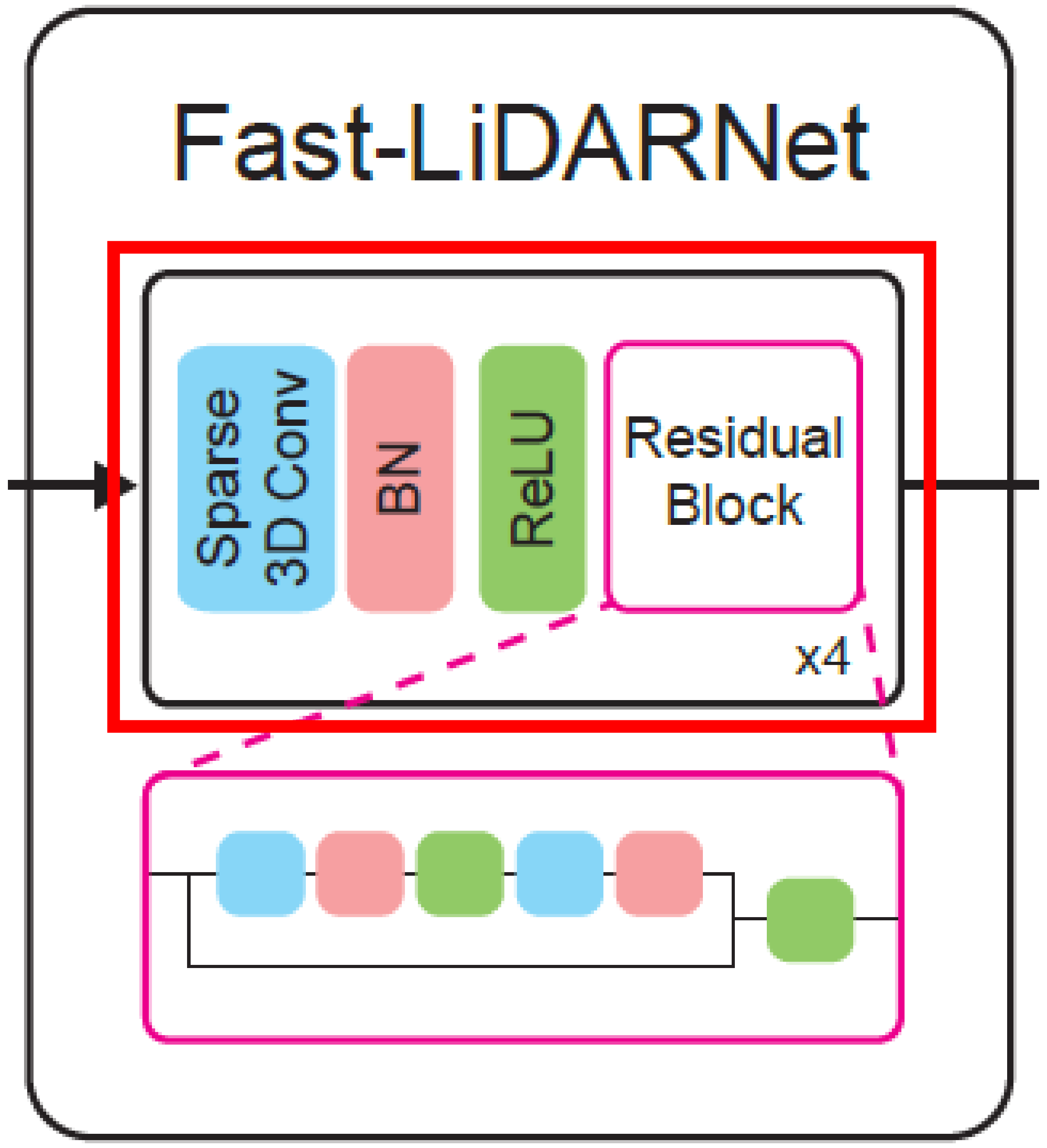
Fast-LiDARNet

2. Redesign of LiDAR processing network

- Bottleneck of 3D sparse convolution
  : memory access rather than computation

- Channel pruning : only effective for computation

- Joint reduction of network size
  : input resolution, channel numbers, network depth

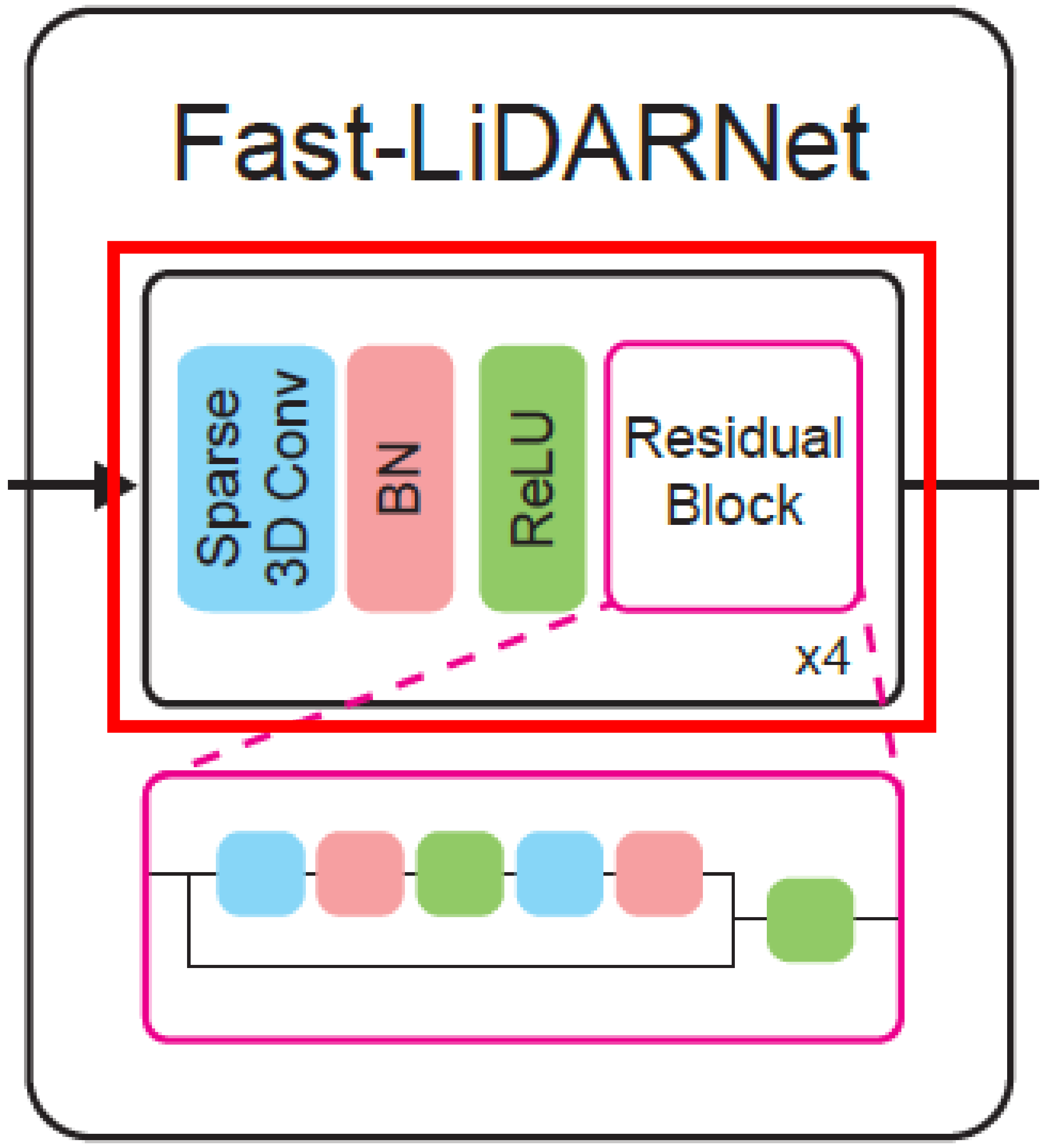


Methodology

Fast-LiDARNet

2. Redesign of LiDAR processing network

- Input resolution = voxel size : 0.2 m
- Network depth = 4 residual block with downsampling layer
- Network channel = 16, 16, 32, 64 channels



Methodology

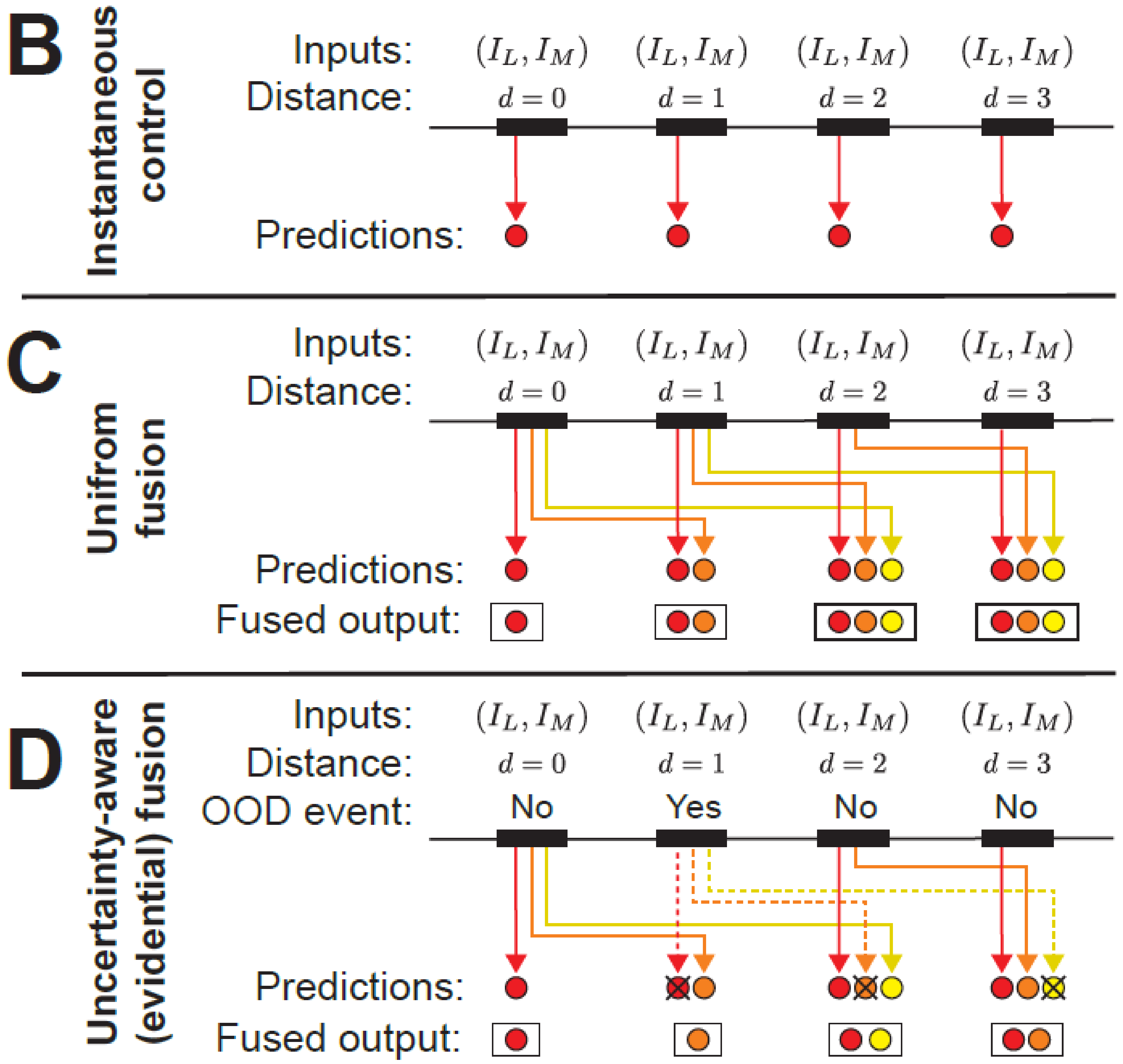
Hybrid Evidential Fusion

1. Uncertainty-aware control

- Use confidence-weighted average of predictions as control input

- The distance, d, is estimated from odometry based on EKF.

- Robust for out-of-distribution event such as sensor failure



Hybrid Evidential Fusion

2. Deep Evidential Regression

$$(y_1, \ldots, y_N) \sim \mathcal{N}(\mu, \sigma^2)$$

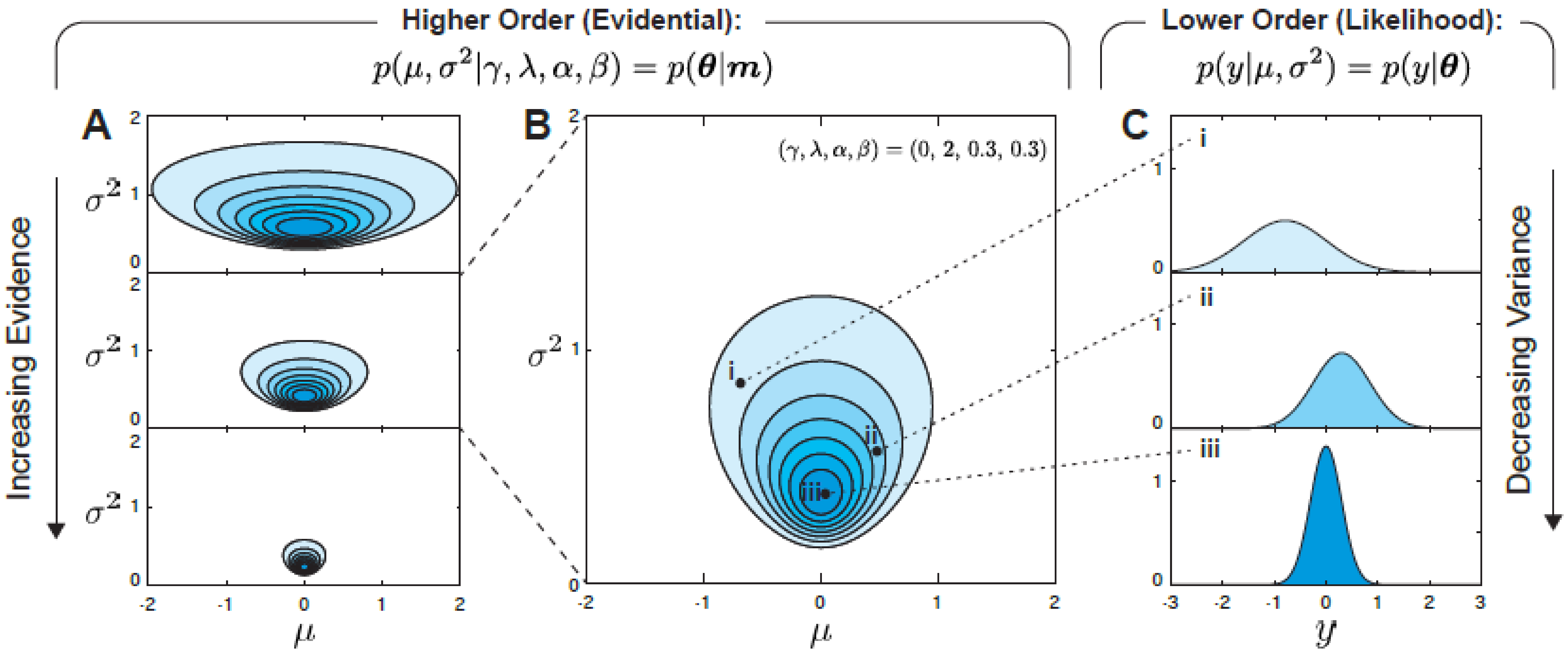
- Observations(yi) are drawn i.i.d from a Gaussian distribution with **unknown mean and variance**

- **Variance of mean = Epistemic uncertainty**

Methodology

Hybrid Evidential Fusion

2. Deep Evidential Regression



Methodology

Hybrid Evidential Fusion

2. Deep Evidential Regression

- To make the prior as Gaussian conjugate prior, place priors over the likelihood variables like below.

$$\mu \sim \mathcal{N}(\gamma, \sigma^2 \upsilon^{-1}) \qquad \sigma^2 \sim \Gamma^{-1}(\alpha, \beta)$$

- Then, our prior(joint distribution) becomes Normal Inverse Gamma (NIG) distribution.

$$p(\underbrace{\mu, \sigma^2}_{\boldsymbol{\theta}} \mid \underbrace{\gamma, \upsilon, \alpha, \beta}_{\boldsymbol{m}}) = \frac{\beta^\alpha \sqrt{\upsilon}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left\{-\frac{2\beta + \upsilon(\gamma - \mu)^2}{2\sigma^2}\right\}$$

Hybrid Evidential Fusion

2. Deep Evidential Regression

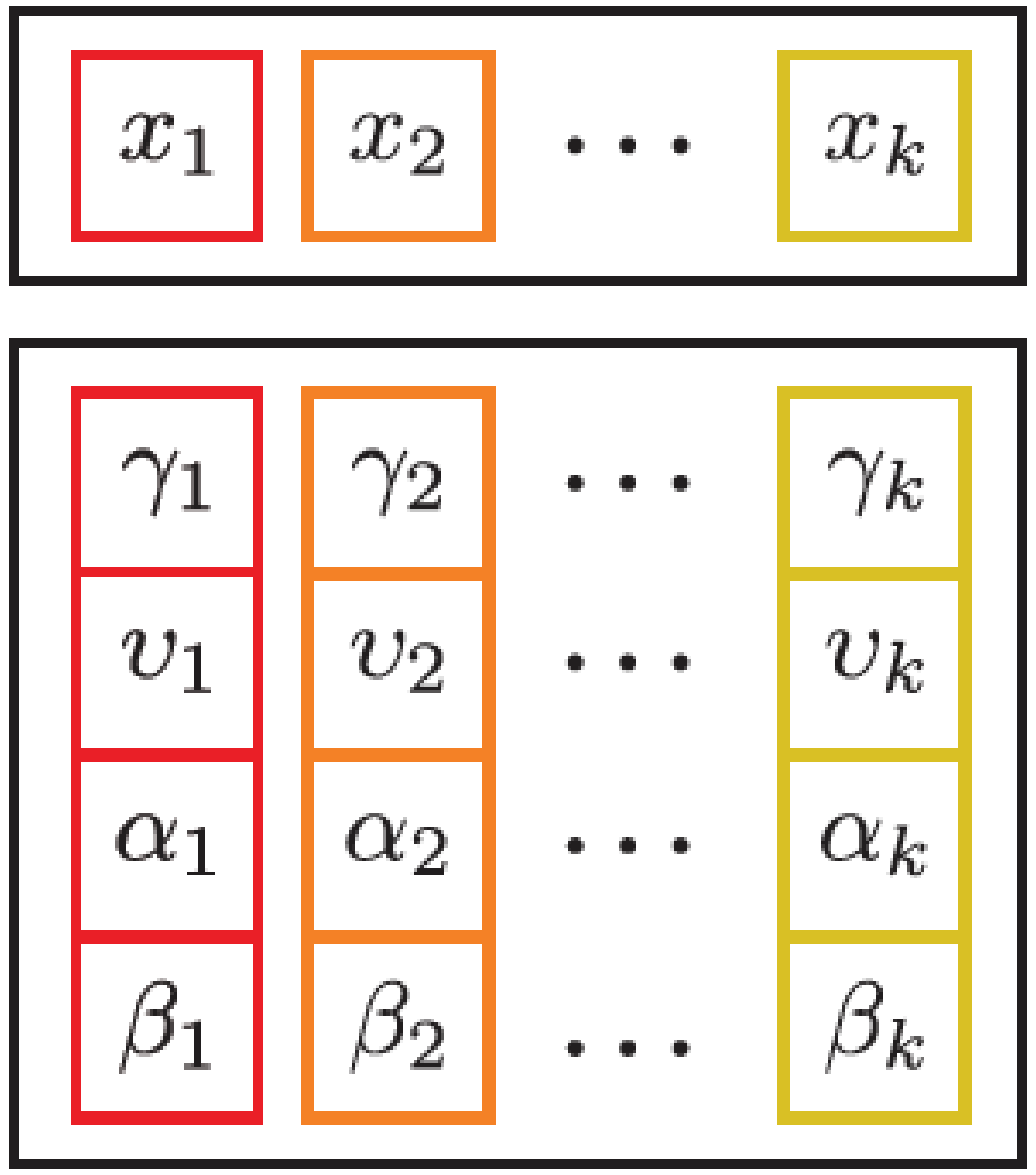
- Loss function

$$\mathcal{L}(\cdot) = \sum\nolimits_k \left(\alpha \mathcal{L}_{\mathrm{MAE}}(\cdot) + \mathcal{L}_{\mathrm{NLL}}(\cdot) + \mathcal{L}_{\mathrm{R}}(\cdot)\right)$$

1) Mean Absolute Error(L1 loss) $\mathcal{L}_{\mathrm{MAE}}(x_k, y_k) = \|x_k - y_k\|_1$

2) Negative Log-Likelihood

3) Evidential Regularizer



Hybrid Evidential Fusion

2. Deep Evidential Regression

- Loss function

  2) Negative Log-Likelihood: Model evidence(aleatoric uncertainty)

$$\mathcal{L}_i^{\mathrm{NLL}} = -\log p(y_i|\boldsymbol{m}) = -\log\left(\mathrm{St}\left(y_i;\gamma,\frac{\beta(1+\upsilon)}{\upsilon\,\alpha},2\alpha\right)\right)$$

$$\mathcal{L}_{\mathrm{NLL}}(\boldsymbol{w}_k,y_k) = \tfrac{1}{2}\log\left(\tfrac{\pi}{\nu_k}\right) - \alpha_k\log(\Omega_k) + (\alpha_k+\tfrac{1}{2})\log\left((y_k-\gamma_k)^2\nu_k+\Omega_k\right) + \log\left(\tfrac{\Gamma(\alpha_k)}{\Gamma(\alpha_k+1/2)}\right)$$

$$\Omega_k = 2\beta_k(1+\nu_k)$$

Hybrid Evidential Fusion

2. Deep Evidential Regression

- Loss function

3) Evidential Regularizer: Minimizing evidence on error(epistemic uncertainty)

$$\mathcal{L}_{\mathrm{R}}(\boldsymbol{w}_k, y_k) = |y_k - \gamma_k| \cdot (2\alpha_k + \nu_k)$$

$$p(\underbrace{\mu, \sigma^2}_{\boldsymbol{\theta}} | \underbrace{\gamma, \upsilon, \alpha, \beta}_{\boldsymbol{m}}) = \frac{\beta^{\alpha}\sqrt{\upsilon}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left\{-\frac{2\beta + \upsilon(\gamma-\mu)^2}{2\sigma^2}\right\}$$

Hybrid Evidential Fusion

3. Uncertainty-aware deployment

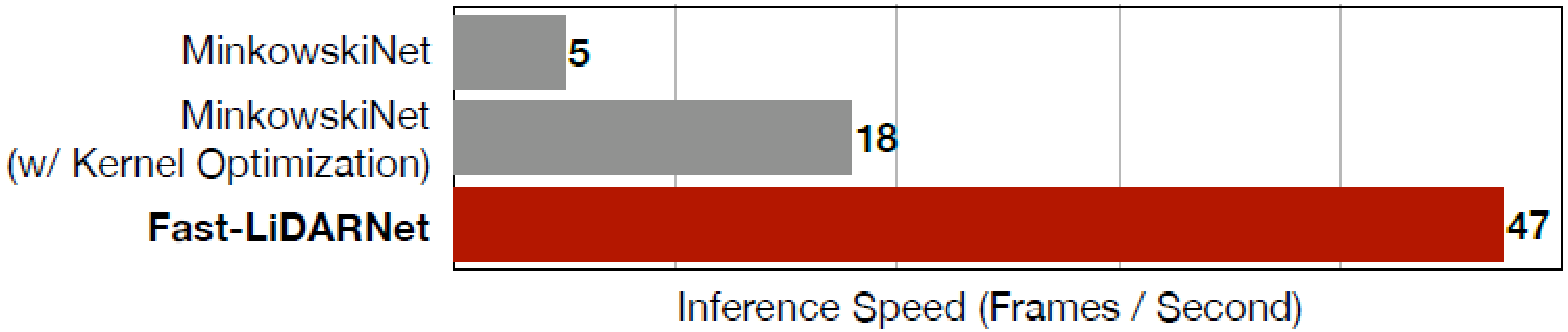
$$\mu \sim \mathcal{N}(\gamma, \sigma^2 \upsilon^{-1}) \qquad \sigma^2 \sim \Gamma^{-1}(\alpha, \beta)$$

- Uncertainty $\mathrm{Var}[\mu_k] \leftarrow \beta_k / (\upsilon_k(\alpha_k - 1))$
- Confidence $\lambda_k \leftarrow 1 \,/\, \mathrm{Var}[\mu_k]$

$$\Lambda^{(d+k)} \leftarrow \Lambda^{(d+k)} \cup \{\lambda_k\}$$
$$\Lambda^{(d)} \leftarrow \Lambda^{(d)} / \textstyle\sum_{\lambda \in \Lambda^{(d)}} \lambda$$

- Evidential fusion $(\sum_j \mathcal{X}_j^{(d)} \Lambda_j^{(d)}) / \|\mathcal{X}^{(d)}\|$

Result

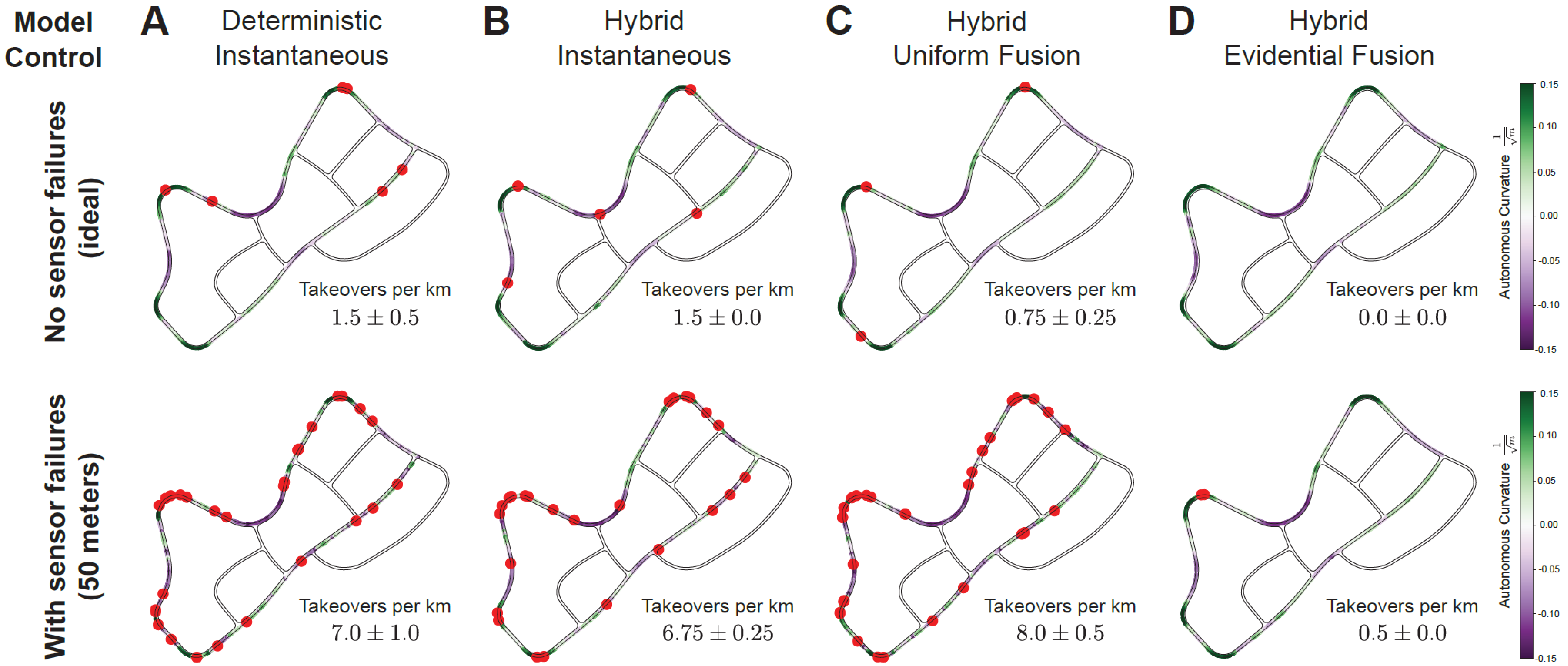


1. Sparse kernel optimization: 3.6x acceleration

2. Model redesign: 2.6x acceleration

=> 47 FPS on GTX 1080Ti / 11 FPS on Jetson AGX Xavier

Result



Model Control
A Deterministic Instantaneous
B Hybrid Instantaneous
C Hybrid Uniform Fusion
D Hybrid Evidential Fusion
No sensor failures (ideal)
With sensor failures (50 meters)
Takeovers per km
1.5 ± 0.5
Takeovers per km
1.5 ± 0.0
Takeovers per km
0.75 ± 0.25
Takeovers per km
0.0 ± 0.0
Takeovers per km
7.0 ± 1.0
Takeovers per km
6.75 ± 0.25
Takeovers per km
8.0 ± 0.5
Takeovers per km
0.5 ± 0.0
Autonomous Curvature
0.15
0.10
0.05
0.00
-0.05
-0.10
-0.15

System setup and Data collection

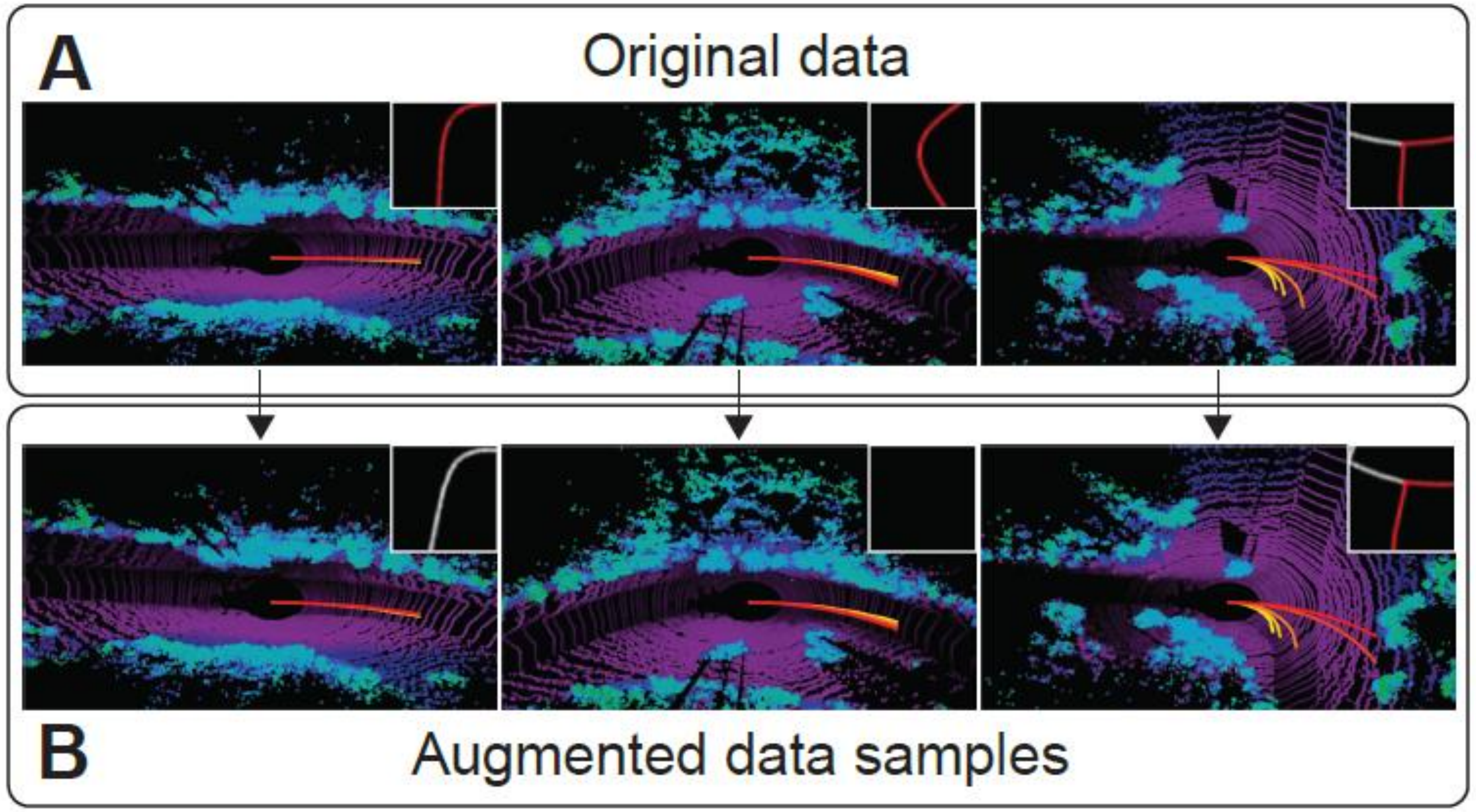
- Velodyne HDL-64E LiDAR operating at 10 Hz
- Coarse-grained GPS data (± 30 m)
- Real-world dataset with 32km of driving takin in a suburban area (29km for training and 3km for testing)
- PC: NVIDIA AGX Pegasus, NVIDIA Jetson Xavier, GeForce GTX 1080 Ti

Data augmentation

1. For point cloud
   - Random scaling by a factor uniformly sampled from [0.95, 1.05]
   - Random rotation by a small yaw angle($\pm 10^{\circ}$)

2. For navigation data
   - Random translation and rotation
   - Black out and absence of the route with probability 0.25

Data augmentation



Experiments

Training

1. Loss function
   : Deep evidential regression(scaled by a factor $1 + \exp[-y_k^2/(2\sigma^2)]$

2. Optimizer
   : ADAM with β_1=0.9, β_2 =0.999 and a weight decay of 10^{-4}

3. Hyperparameters
   : Epochs = 250, Batch-size = 64, learning rate = 0.003 with cosine decay schedule

Limitation

1. Is it generalizable?
   : The experiment is only done on trained road.

2. How about urban area?
   : The experimented area was suburban area where the traffic sign or obstacles are not exist. Thus, the complexity of the setting should be more tough.

3. The Hybrid Evidential Fusion can only deal with limited out-of distribution event.
   : What if something suddenly jumps into the road?

UNIST

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

2007

THANK YOU